Multi-Disciplinary Integrated Design Automation Tool for Automotive Cyber-Physical Systems

Arquimedes Canedo[†], Mohammad Abdullah Al Faruque[‡], Jan H. Richter[§]
[†]Siemens Corporation, Corporate Technology, Princeton, USA
[‡]Department of Electrical Engineering and Computer Science, University of California Irvine, Irvine, US
[§]Siemens AG, I IA ATS 4, Nuremberg, Germany

***Abstract*—This paper presents our multi-year experience in the development of a Functional Modeling Compiler (FMC), a new model-based design tool for the development of multi-disciplinary automotive cyber-physical systems. We show how system-level simulation models suitable for design-space exploration of complex architectures can be synthesized from functional specifications to test and validate the interactions between ECUs, control algorithms, and the multi-physics.**

I. Introduction

The current automotive design process is complex and multi-disciplinary. The design is passed hundreds of times through hundreds of personnel from various organizations from the initial concept through construction and road test. The dense integration of embedded systems[1] with physical processes in a car is critical to achieve good fuel economy, low emissions, and better safety, among others. Currently, automotive design has been siloed into specific disciplines and supported by highly specialized but domain-specific model-based design (MBD) automation tools. For example, mechanical engineering is done in computer-aided design (CAD) and engineering (CAE) tools; electrical engineering is done in electronic design automation (EDA) and wire harness design tools; control engineering is done in Matlab/Simulink and Modelica; and software engineering is done in UML and in-house software development environments.

This paper reports our experience in the development of a novel multi-disciplinary integrated design automation tool for automotive cyber-physical systems. Our tool shows that the various disciplines in automotive design can be brought together to enhance the communication and requirements negotiation among engineers and organizations, enable multi-disciplinary simulations to evaluate the system-level impact of domain-specific design decisions, and reduce the overall design cycle. Our method relies on functional modeling to create a technology-independent description of *what the system does*, and uses a Functional Modeling Compiler (FMC) to synthesize technology-dependent solutions that can be directly used in multi-disciplinary simulations for validation and design-space exploration.

II. Functional Modeling Compiler for Component-based Simulation

Automotive engineering consists of multiple design iterations starting at concept design where the requirements of the product are defined, to detail design where the physical system is realized and tested. Functional modeling is a design activity where the informal or semi-formal requirements are formally specified in terms of *functions*. A function describes *what the system does* in terms of energy, material, and signal transformations yet it remains technology independent. Functional models are written in the Functional Basis [2], a high-level language close to natural language with well defined syntax and semantics to facilitate inter-disciplinary communication among engineers from different domains. For example, the function "`convert chemical energy to rotational mechanical energy`" implies the use of fuel (chemical energy) to generate torque (rotational mechanical energy) but does not specify whether it is a gasoline, gas, or diesel car with a four-stroke or a Wankel cycle. The Functional Basis taxonomy provides 32 elementary functions and 18 flow types that can be used to compose more complex functions.

In the existing workflow, the functional model is a static document used by the domain engineers to translate requirements into engineering specifications in each of the disciplines that allocate functions to actual components. Unfortunately, it is very difficult for computer-aided design tools from different disciplines to exchange data and take advantage of the model-based design at the system-level. To overcome the limitations of the current siloed development, this paper presents a FMC capable of automating the allocation of functions to components and generating feasible multi-disciplinary simulation models to validate different architectures and various components (e.g. ECUs, transmissions, engines, etc.) at the system-level. Our FMC is intended to be used as a design-space exploration tool at the concept design phase to evaluate how different combinations of automotive components can be leveraged to achieve the functional and non-functional[2] requirements.

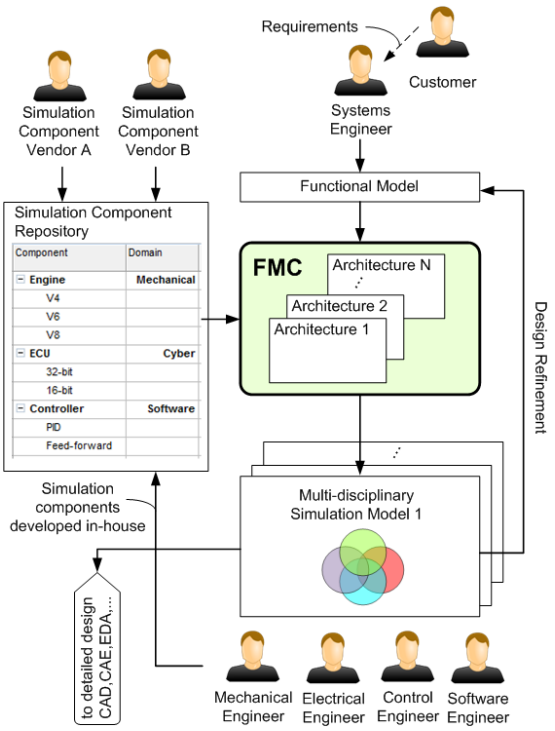

Fig. 1. FMC allocates simulation components to functions in a functional model. The result are multi-disciplinary simulation models that domain-engineers can use during concept design to validate new components in existing architectures or innovative and new architectures.

Our FMC leverages the technological advances in hybrid simulation languages such as AMESim [3] and Modelica [4] to generate multi-disciplinary simulation models to validate the interactions between the physical and the cyber components of a car. The commercial support of simulation component libraries written in these languages allows us to generate high-fidelity functional simulation models with components that have been validated by the vendors and are very close to the behavior of their real-life counterpart. These components include multi-physics components, and ECUs and control algorithms for internal combustion engines, automatic gearboxes, fuel cells, series- and parallel-electric cars. Figure 1 shows how the FMC is used to perform a design-space exploration of

[1]Automotive embedded systems are often referred to as Electronic Control Units or ECUs. A modern car contains about 70 ECUs.

[2]Non-functional requirements refer to performance characteristics of a car such as acceleration, noise-vibration-harshness (NVH), etc.

978-3-9815370-2-4/DATE14/©2014 EDAA

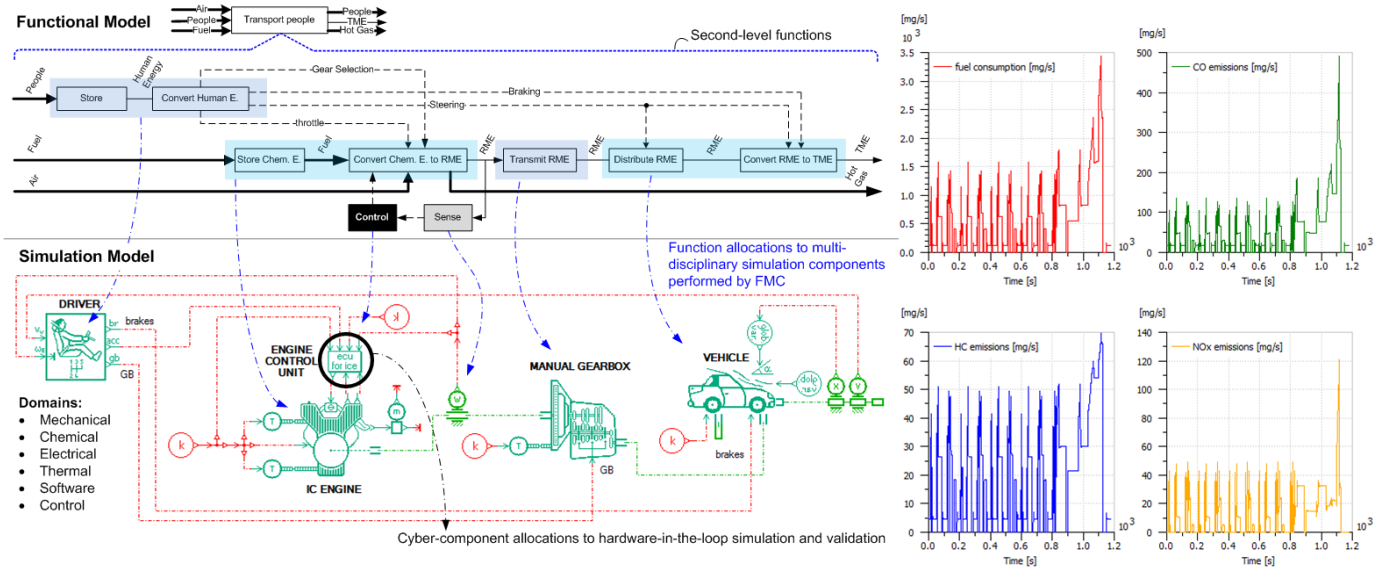

Fig. 2. Functional model of an internal combustion engine car showing the allocation to simulation components performed by the FMC. This functional decomposition shows two levels and illustrates the embodiment-independent view of a system that is capable of maintaining variability of different architectures (e.g. gasoline, gas, or diesel engines).

different architectures by combining different simulation components (e.g. engines, ECUs, controllers, etc.) stored and classified (e.g. mechanical, cyber, software, etc.) in a simulation component repository. The input to the FMC is a functional model and it generates multi-disciplinary simulation models as an output[3].

III. Automotive Architectures Synthesis Example

Figure 2 shows a typical functional model and functional decomposition of an internal combustion engine car. The top-level function "`transport people`" describes the purpose of a "Car". It inputs three material flows (bold arrows for Air, People, and Fuel) and outputs two material flows (People and Hot Gas) and one energy flow (arrow representing translational mechanical energy or TME). The second-level decomposition shows the functions and flows representing the functionality of the combustion engine, transmission, driveline, wheels, and fuel subsystems. For example, the function "`convert chemical energy to rotational mechanical energy`" receives two material flows (Fuel and Air) and one signal flow (throttle) and produces RME and Hot Gas as outputs. Using natural language, these functions remain technology-independent and allow non experts to model the environment to design their subsystems. For example, this particular model was created by a software engineer with marginal understanding of mechanical engineering principles who wanted to design the engine control system (ECU and its controller software) represented by the "`Sense`" function (gray box) and the "`Control`" function (black-box). FMC simplifies the design process by allocating functions to high-fidelity simulation components and creating a multi-disciplinary simulation model that domain engineers can use to design and validate their subsystems in software- and hardware-in-the-loop simulations (SILS, HILS). In this example, the effects of the ECU on fuel consumption, CO, HC, and NOx emissions can be tested and validated as shown in Figure 3 plots.

Notice that with the help of FMC, the software engineer can simply change the "Fuel" flow for an "Electrical Energy" flow in the functional model and automatically synthesize an electric vehicle simulation model (See Figure 3) that, instead of a combustion engine, it includes a battery, an electric motor, and a suitable ECU to control these electric components. The plot in Figure 3 shows the validation of the "torque command" control signals provided by the ECU to the electric motor and the motor behavior or "motor torque" accomplished with this controller. Currently, our FMC supports functional models written using the NIST Functional Basis [2] and synthesizes multi-disciplinary simulation models using the AMESim component library and Modelica Standard Library.

[3]The details of our mapping algorithm that allocates components to functions according to a context-based analysis can be found in [1].

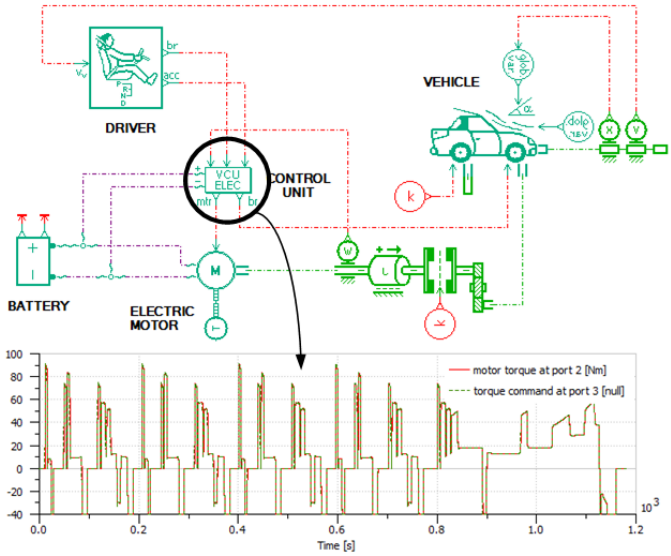

Fig. 3. Synthesized electric vehicle architecture from a functional model using the FMC and AMESim components.

IV. Summary

This paper highlights our research activities in developing the Functional Modeling Compiler, a new model-based design automation tool for the concept design of cyber-physical systems. The FMC synthesizes multi-disciplinary simulation models in AMESim and Modelica from functional information. FMC enables embedded software engineers to automatically synthesize multiple vehicle architectures ready for SILS and HILS and validate new ECUs and control strategies for existing and new architectures (e.g. hybrid and electric vehicles). Because embedded software may be auto-generated from such simulation models using existing tools, we strongly advocate the high-level synthesis of future automotive CPS in this project.

References

[1] A. Canedo, E. Schwarzenbach, and M. A. Al Faruque. Context-sensitive synthesis of executable functional models of cyber-physical systems. In *Proceedings of the ACM/IEEE 4th International Conference on Cyber-Physical Systems*, ICCPS '13, pages 99–108, 2013.

[2] J. Hirtz, R. B. Stone, S. Szykman, D. A. McAdams, and K. L. Wood. A functional basis for engineering design: Reconciling and evolving previous efforts. Technical report, NIST, 2002.

[3] LMS Imagine.Lab AMESim. http://www.lmsintl.com/.

[4] Modelica Association, Modelica. https://modelica.org/.